Information Fusion 99 (2023) 101899

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full length article

Effective sample pairs based contrastive learning for clustering

Jun Yin [a], Haowei Wu [a], Shiliang Sun [b,c,*]

[a] College of Information Engineering, Shanghai Maritime University, Shanghai, 201306, China
[b] School of Computer Science and Technology, East China Normal University, Shanghai, 200062, China
[c] Key Laboratory of Advanced Theory and Application in Statistics and Data Science, Ministry of Education, Shanghai, 200062, China

ARTICLE INFO

Keywords:
Representation learning
Contrastive learning
Deep clustering
Nearest neighbor

ABSTRACT

As an indispensable branch of unsupervised learning, deep clustering is rapidly emerging along with the growth of deep neural networks. Recently, contrastive learning paradigm has been combined with deep clustering to achieve more competitive performance. However, previous works mostly employ random augmentations to construct sample pairs for contrastive clustering. Different augmentations of a sample are treated as positive sample pairs, which may result in false positives and ignore the semantic variations of different samples. To address these limitations, we present a novel end-to-end contrastive clustering framework termed Contrastive Clustering with Effective Sample pairs construction (CCES), which obtains more semantic information by jointly leveraging an effective data augmentation method ContrastiveCrop and constructing positive sample pairs based on nearest-neighbor mining. Specifically, we augment original samples by adopting ContrastiveCrop, which explicitly reduces false positives and enlarges the variance of samples. Further, with the extracted feature representations, we provide a strategy to construct positive sample pairs via a sample and its nearest neighbor for instance-wise and cluster-wise contrastive learning. Experimental results on four challenging datasets demonstrate the effectiveness of CCES for clustering, which surpasses the state-of-the-art deep clustering methods.

1. Introduction

In the past few decades, unsupervised learning has attracted increasing attention and shown its feasibility in the field of pattern recognition and computer vision. As a research hotspot of unsupervised learning, clustering seeks to gather similar samples in one cluster and partition dissimilar samples into different clusters.

Traditional clustering methods, such as K-means [1], Spectral Clustering (SC) [2], and Agglomerative Clustering (AC) [3], typically focus on low-dimensional information, which may lead to undesirable performance on enormous and noisy datasets. With the development of deep learning [4], clustering tasks coupled with representation learning have reached more competitive results. As one of the earliest deep clustering approach, Deep Embedded Clustering (DEC) [5] attempts to incorporate feature representation learning with clustering into a deep neural network. Joint Unsupervised LEarning (JULE) [6] uses CNN to extract feature representations and improve the results of clustering, which in turn provides a supervised signal for feature representation. Analogously, Deep Clustering [7] uses k-means to partition the features extracted from the network and predicts cluster assignments by a discriminative loss, so that parameters of the network can be updated. PartItion Confidence mAximisation (PICA) [8] considers the cluster-level distributions of all samples, so that it can get the global information to train the model. Despite the progress made by the aforementioned methods, they usually utilize the entire distribution to conduct representation learning and clustering, ignoring instance-wise relationships. Moreover, most of them take an offline training manner, limiting the scenarios of online clustering for large scale data.

Recently, contrastive learning [9,10] gradually show its capability in deep clustering. Li et al. [11] proposed Contrastive Clustering(CC), which extends the contrastive learning framework to clustering tasks and exploits instance- and cluster-level contrastive learning to obtain outstanding performance. Van Gansbeke et al. [12] proposed a method named SCAN (Semantic Clustering by Adopting Nearest neighbors) to employ contrastive learning and consider semantic information by exploring nearest neighbors in a two-stage manner. After that, lots of works [13,14] try to integrate contrastive learning into deep clustering tasks. Fig. 1 illustrates the mainstream deep clustering methods based on contrastive learning, which is mainly categorized into two types according to the training strategies. The one is one-stage end-to-end training, such as CC, and the other one is multi-stage training, such as SCAN.

Although the performance of the mentioned contrastive clustering methods has been greatly improved, they pay little attention to semantic information. On one hand, almost all the existing contrastive

* Corresponding author.
E-mail addresses: junyin@shmtu.edu.cn (J. Yin), 202130310128@stu.shmtu.edu.cn (H. Wu), slsun@cs.ecnu.edu.cn (S. Sun).

https://doi.org/10.1016/j.inffus.2023.101899
Received 23 January 2023; Received in revised form 4 June 2023; Accepted 23 June 2023
Available online 28 June 2023

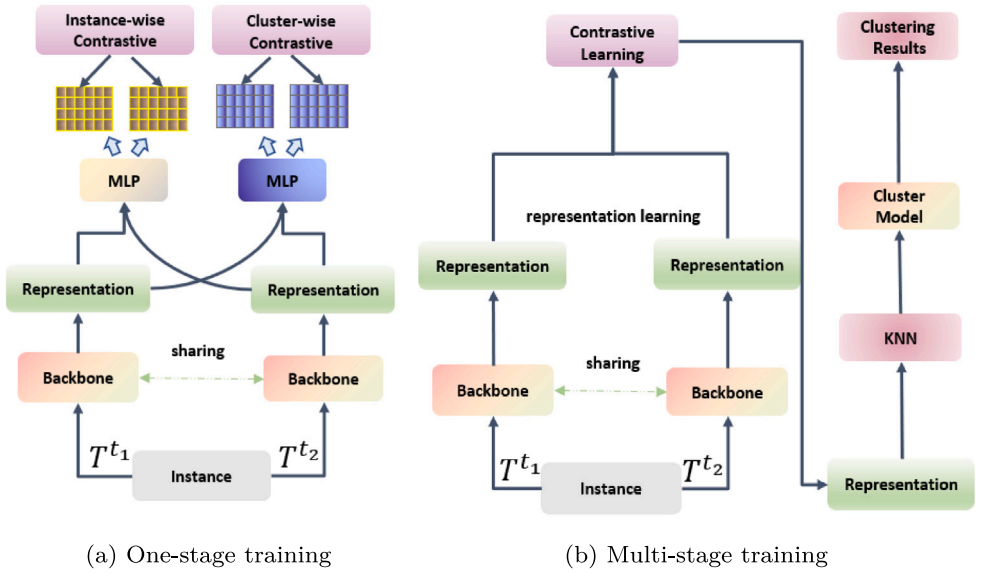

Fig. 1. Different training frameworks of contrastive clustering. (a) An end-to-end contrastive clustering framework (b) A multi-stage contrastive clustering framework.

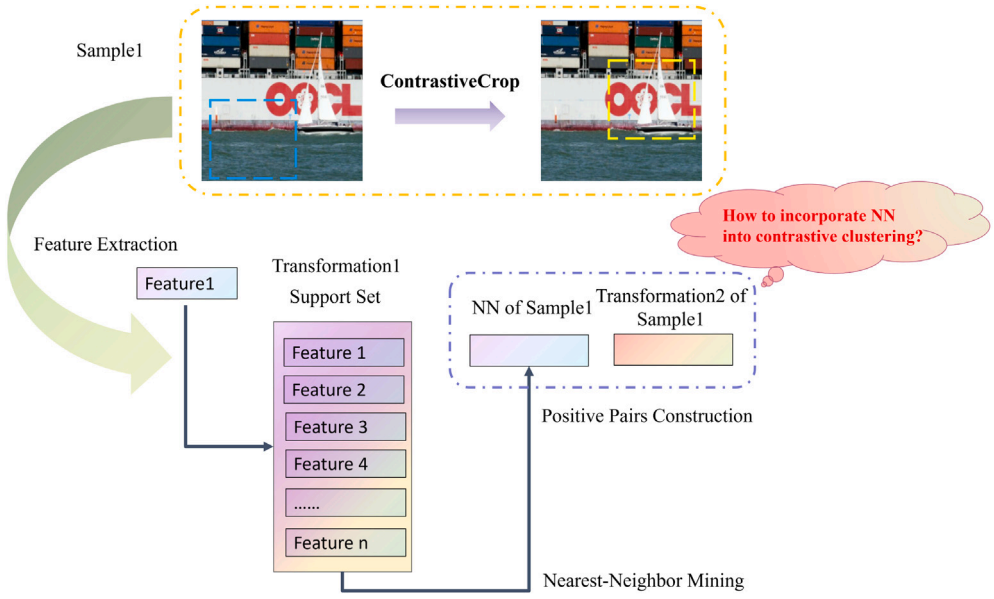

Fig. 2. Visualization of the ContrastiveCrop and the nearest-neighbor mining. The blue dashed box indicates RandomCrop and the yellow dashed box indicates ContrastiveCrop. Obviously, the latter can locate target of the image. Further, we provides a novel way of constructing positive sample pairs via nearest-neighbor mining.

clustering methods utilize ordinary data augmentations (e.g. RandomCrop), which may obtain the background of original samples. On the other hand, positive sample pairs are usually derived from data augmentations of the same image, and their semantic variation is not sufficient. More recently, Peng et al. [15] intended to craft more semantic contrastive views with promising augmentations ContrastiveCrop. Dwibedi et al. [16] denoted that nearest neighbor based on support set could substantially boost the performance of contrastive learning. However, both of them are designed for the general contrastive learning and also lack the ability of directly learning the clustering structure. Thus, how to integrate effective augmentations and nearest-neighbor mining into the contrastive clustering framework to extract more semantic information is still a problem. Some multi-stage training methods [12, 17] have been proposed to obtain more semantic information, which usually explore the nearest neighbor after representation learning, but they cannot realize an end-to end clustering. Moreover, multi-stage training pipeline is often affected by the initial value of weights in representation learning, and error propagation occurs frequently.

To overcome the above shortcomings, we propose a novel contrastive clustering method, termed Contrastive Clustering with Effective Sample pairs construction (CCES). As shown in Fig. 2, we employ ContrastiveCrop, which considers more semantic information between instances. More suitable crops are generated by ContrastiveCrop, which can reduce false positives and enlarge the variance between sample pairs. Further, in order to explore richer semantic information and reduce the complexity of the model, we introduce a support set for nearest-neighbor mining, construct positive sample pairs via a sample and its nearest neighbor, and introduce them into an online clustering framework. Experimental results show the superiority of the proposed CCES. Our contributions are summarized as follows:

(1) We review the existing deep clustering frameworks based on contrastive learning and reveal the weakness of previous methods. To address these issues, we present a novel contrastive

clustering method termed CCES, which conducts instance-wise and cluster-wise contrastive learning simultaneously.

(2) Our work deviates from the previous contrastive clustering methods, which either ignores semantic information or discards end-to-end training. To the best of our knowledge, CCES is the first work, which innovatively conjoins the benefits of both the effective data augmentation method ContrastiveCrop and the nearest-neighbor mining for the task of clustering. Besides, CCES can be realized in an end-to-end and online clustering fashion.

(3) The superiority of the proposed CCES is confirmed by extensive experiments as well as rigorous ablation studies. Compared to many competitive works, CCES consistently achieves superior performances. It obtains 4% higher accuracy than previous baseline the Tiny-ImageNet dataset.

2. Related works

2.1. Contrastive learning

In recent years, as a subset of unsupervised representation learning, contrastive learning [9,10] plays a key role in unsupervised representation learning. Contrastive learning aims to map original samples to a subspace by minimizing the distance between positive sample pairs and maximizing the distance between negative sample pairs. In 2006, Le Cun et al. [18] proposed contrastive loss, which promotes the development of contrastive learning. Noise-Contrastive Estimation (NCE) [19] is a statistical model estimation method, which transforms the contrastive learning problem into a binary classification problem. As a variant of NCE, InfoNCE [20] transforms the problem into multi-class classification, which is adopted in Contrastive Predictive Coding (CPC). It realizes the purpose of predicting the future by maximizing the mutual information between current state features and future input features, allowing the model to learn more friendly feature representations. MoCo [10] considers that contrastive learning ought to be regarded as a process of searching dictionaries. It introduces queues to store more negative samples to boost the performance. SimCLR [9] mainly depends on large batch size and data augmentations. Meanwhile, it employs the non-linear layer into the contrastive learning framework to improve the effectiveness of representation learning. After that, a broad family of works attempt to construct more reasonable sample pairs. From the perspective of data augmentations, Wang et al. [21] and Zheng et al. [22] utilized strong augmentation and weak augmentation respectively to optimize the framework of contrastive learning. Robinson et al. [23], Kalantidis et al. [24] and Zhu et al. [25] applied more hard negatives in contrastive learning to learn more discriminative representations.

2.2. Deep clustering

Deep clustering has been proven to be more effective than traditional clustering because of the strong feature extraction capability. The structure of model is the main differences of different deep clustering methods. Ji et al. [26] constructed DSC-Nets by introducing a self-expressive layer between the encoder and decoder to tackle the subspace clustering problem. Jiang et al. [27] introduced Variational Deep Embedding (VaDE) that optimizes the model by maximizing the evidence lower bound. Recently, some deep clustering methods emerged that discard the reconstruction loss, directly construct clustering loss. Xie et al. [5] projected data points from a high dimensional space into a low dimensional space to minimize a clustering loss. Wu et al. [28] proposed Deep Comprehensive Correlation Mining (DCCM) to explore correlation behind the unlabeled data in different aspects. Guo et al. [29] developed a novel algorithm that combines data augmentation with self-paced learning to obtain robust clustering features. Ji et al. [30] proposed Invariant Information Clustering (IIC), which exploits mutual information to realize label prediction and works in an end-to-end manner.

Subsequently, many deep clustering methods are combined with contrastive learning to substantially boost deep clustering performance. According to different training manners, contrastive clustering can be mainly divided into two basic categories, i.e., end-to-end and multi-stage. As the end-to-end training strategy, CC [11] first considers the label as representation, and performs contrastive clustering in the instance level and cluster level. Similarly, Doubly Contrastive Deep Clustering (DCDC) [13] constructs contrastive loss from sample view and class view, which could learn more discriminative representations. Deep Robust Clustering (DRC) [14] utilizes assignment probability and assignment feature to increase inter-class diversities and decrease intra-class diversities simultaneously. Strongly Augmented Contrastive Clustering (SACC) [31] introduces strong data augmentations, which extends contrastive clustering from binary branching structures to ternary branching structures. As the multi-stage training strategy, SCAN [12] pretrains an unsupervised model with the idea of contrastive learning, then fixes parameters of pretrained model and explores nearest neighbors to perform deep clustering. Different from SCAN, Nearest Neighbor Matching (NNM) [32] matches instances with their nearest neighbors from the perspective of local and global information respectively. Contrastive Learning and K-Nearest Neighbors (CLKNN) [17] trains the model with contrastive loss in representation learning stage, and explores nearest neighbors of each instance in the clustering stage. Nearest Neighbor Contrastive Clustering (NNCC) [33] combines contrastive learning with neighbor relation mining, which are updated alternately during training. Semantic Pseudo-labeling-based Image ClustEring (SPICE) [34] introduces semantic pseudo-labeling for image clustering, which divides the clustering network into a feature model for measuring the instance-level similarity and a clustering head for identifying the cluster-level discrepancy.

The aforementioned contrastive clustering methods typically employ RandomCrop as a data augmentation technique. However, this technique may cut out the background of the image and overlook the primary object, which makes the background be incorrectly considered as a positive sample. To address this issue, in the proposed CCES, we incorporate ContrastiveCrop as a data augmentation technique to precisely locate the object in the image, which can reduce the probability of the occurrence of the false positives. Besides, in the above contrastive clustering methods, positive sample pairs are only constructed via two different augmentations of a sample, and the similarity of positive sample pair is maximized. They neglect the intra-class variations and discard semantic information between different samples. In order to capture more semantic-level information, in this paper, we construct positive sample pairs not only by different augmentations of a sample but also similar samples. To achieve this goal, we integrate the nearest-neighbor mining into an end-to-end training framework, constructing positive sample pairs via augmented samples and their nearest neighbors.

3. Approach

In this section, we briefly describe a contrastive learning framework SimCLR, which is closely related to our method. Then, we propose our end-to-end contrastive clustering framework CCES, which concentrates on an effective data augmentation method ContrastiveCrop and a novel way of constructing positive sample pairs via the nearest neighbor to obtain more semantic information for instance-wise and cluster-wise contrastive learning. Finally, we give the loss function of CCES.

3.1. Contrastive learning framework

Contrastive learning is a type of self-supervised learning that learns knowledge from unlabeled samples. It pulls similar sample pairs closer

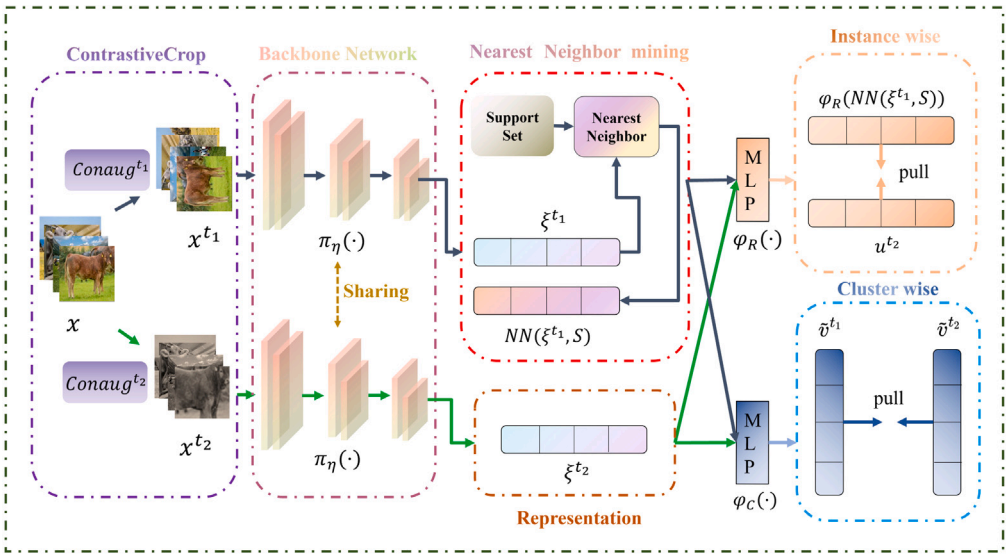

Fig. 3. Overview of CCES. By successfully integrating ContrastiveCrop and support set into the proposed framework, a siamese neural network with better crops are applied to learn more semantic feature representations. Then, these representations are stored in the support set for the nearest-neighbor mining, which will be fed to the projector heads $\varphi_R(\cdot)$ and $\varphi_C(\cdot)$ to realize instance-wise and cluster-wise contrastive learning in an end-to-end fashion.

while pushes dissimilar sample pairs away via a contrastive loss. Recently, the development of contrastive learning has provided a solid foundation for deep clustering. Here, we describe a mainstream contrastive learning framework SimCLR. In the proposed CCES, we employ the SimCLR framework.

SimCLR constructs positive sample pairs and negative sample pairs by augmentations, and then utilizes a weight-sharing network model with upper and lower branches to realize contrastive learning. The contribution of SimCLR is the adoption of the larger batch size, effective data augmentations, and the nonlinear projection layers.

For the ith sample, SimCLR generates augmented samples $x_i^{t_1}$ and $x_i^{t_2}$ from two different augmentations. Further, neural network transforms augmented samples $x_i^{t_1}$ and $x_i^{t_2}$ to feature representations $u_i^{t_1}$ and $u_i^{t_2}$, which is regarded as a positive sample pair. The loss function of the augmented sample $x_i^{t_1}$ is defined as

$$\ell_i^{t_1} = -\log \frac{\exp\left(s\left(u_i^{t_1}, u_i^{t_2}\right)/\tau\right)}{\sum_{j=1}^{N} \mathbf{1}_{[j\neq i]}\left[\exp\left(s\left(u_i^{t_1}, u_j^{t_1}\right)/\tau\right) + \exp\left(s\left(u_i^{t_1}, u_j^{t_2}\right)/\tau\right)\right]}, \quad (1)$$

where N is the batch size, τ is the temperature parameter and $s(\cdot)$ is a similarity function.

3.2. The framework of CCES

In this paper, we present an end-to-end contrastive clustering framework CCES. Fig. 3 illustrates the proposed CCES framework. From Fig. 3, we can see that CCES jointly leverages an effective data augmentation method ContrastiveCrop and a novel way of constructing positive sample pairs based on nearest-neighbor mining.

CCES first augments the original sample x_i with ContrastiveCrop $Conaug(\cdot)$ to generate two dependent samples, denoted as $x_i^{t_1} = Conaug^{t_1}(x_i)$ and $x_i^{t_2} = Conaug^{t_2}(x_i)$. Then, the backbone network $\pi_\eta(\cdot)$ transforms augmented samples $x_i^{t_1}$ and $x_i^{t_2}$ to feature representations $\xi_i^{t_1}$ and $\xi_i^{t_2}$ respectively. Furthermore, CCES applies the support set S for the nearest-neighbor mining. It gets the nearest neighbor of $\xi_i^{t_1}$ from the support set, named $NN\left(\xi_i^{t_1}, S\right)$, and feeds $NN\left(\xi_i^{t_1}, S\right)$ to the instance-wise MLP $\varphi_R(\cdot)$ and cluster-wise MLP $\varphi_C(\cdot)$ to obtain the nearest-neighbor features $\varphi_R\left(NN\left(\xi_i^{t_1}, S\right)\right)$ and $\varphi_C\left(NN\left(\xi_i^{t_1}, S\right)\right)$. Simultaneously, CCES feeds $\xi_i^{t_2}$ to $\varphi_R(\cdot)$ and $\varphi_C(\cdot)$ directly to obtain feature representations $u_i^{t_2}$ and $v_i^{t_2}$. Finally, CCES minimizes instance-wise contrastive loss in row space and minimizes cluster-wise contrastive loss in column space.

In CCES, feature extraction and clustering are integrated into an end-to-end training framework through performing instance-wise and cluster-wise contrastive learning simultaneously. In multi-stage contrastive clustering methods, features are firstly extracted by contrastive learning and then the clustering method (e.g. K-means) is performed to obtain the clustering results. These approaches separate feature extraction and clustering, which may make the extracted features unsuitable for clustering task. However, the end-to-end learning mechanism adopted in CCES can reduce inter-cluster similarities, providing feature representations that favor clustering.

3.3. Data augmentation and nearest neighbor mining in CCES

Previous contrastive learning methods commonly adopt RandomCrop, which may crop the background of the image and regard them as positives. These crops are false positives actually. To avoid false positives as much as possible, CCES employs ContrastiveCrop for data augmentation. ContrastiveCrop firstly obtains bounding box which contains the target of the image. Then the bounding box is cropped, and the obtained crops almost contain some parts of the target. Thus the false positives can be avoided. Considering that the target is near the center of the image, an easy way to get the bounding box is to take the center point of the image as a starting point, and extend outwards to get a rectangular box. By jointly leveraging ContrastiveCrop and previous augmentations (ColorJitter, RandomGrayscale, etc.), feature representations with more semantic information are used in the clustering task to improve the performance of CCES.

Previous contrastive clustering methods almost utilize different augmentations to construct positive sample pairs, which often leads to misjudgment of different samples belonging to the same cluster and ignore semantic information. To solve this problem, we introduce the nearest neighbor, which acts as small semantic perturbations in the latent space and explores richer semantic variations. To realize this, in CCES, we adopt the support set S for the nearest-neighbor mining in the contrastive clustering framework, minimizing the distance between an augmented sample and its nearest neighbor. Specifically, we mine the nearest neighbor of $\xi_i^{t_1}$ through $NN\left(\xi_i^{t_1}, S\right) = \arg\min_{s\in S} \left\|\xi_i^{t_1} - s\right\|$, where $\|\cdot\|$ is the ℓ_2-norm. Then, feature representations of $NN\left(\xi_i^{t_1}, S\right)$ are fed to $\varphi_R(\cdot)$ and $\varphi_C(\cdot)$ to obtain the low-dimensional nearest-neighbor

features $\varphi_R\left(NN\left(\xi_i^{t_1}, S\right)\right)$ and $\varphi_C\left(NN\left(\xi_i^{t_1}, S\right)\right)$, respectively. Different from previous works, we construct positive sample pair with the feature $u_i^{t_2}$ and its neighbor feature $\varphi_R\left(NN\left(\xi_i^{t_1}, S\right)\right)$ for instance-wise contrastive learning. Simultaneously, we construct positive sample pair with the feature $v_i^{t_2}$ and its nearest neighbor feature $\varphi_C\left(NN\left(\xi_i^{t_1}, S\right)\right)$ for cluster-wise contrastive learning. The support set S is actually a queue, satisfying the first-in-first-out principle, which is generated by random initialization. The support set S is updated in each training epoch. In the update, the first n (batch size) elements in the queue are discarded and n new representations enter the end of the queue.

3.4. CCES loss

The loss function of CCES consists of two parts, namely the instance-wise loss and the cluster-wise loss. We leverage the cosine similarity function to measure the similarity between sample pairs, which is defined as

$$s(z_1, z_2) = \frac{z_1 \cdot z_2^{\top}}{\|z_1\|\|z_2\|}, \tag{2}$$

where z_1 and z_2 are row vectors. For instance-wise contrastive loss, we maximize the agreement of positive sample pairs employed from augmented samples and their nearest neighbors while minimizing the agreement of negative sample pairs. The instance-wise contrastive loss from augmentation t_1 is formulated as

$$\ell_i^{t_1} = -\log \frac{\exp\left(s\left(\varphi_R\left(NN\left(\xi_i^{t_1}, s\right)\right), u_i^{t_2}\right)/\tau_R\right)}{\sum_{j=1}^{N} \mathbf{1}_{[j\neq i]}\left[\begin{array}{l}\exp\left(s\left(\varphi_R\left(NN\left(\xi_i^{t_1}, s\right)\right), u_j^{t_1}\right)/\tau_R\right) \\ +\exp\left(s\left(\varphi_R\left(NN\left(\xi_i^{t_1}, s\right)\right), u_j^{t_2}\right)/\tau_R\right)\end{array}\right]}, \tag{3}$$

where τ_R is the instance-wise temperature parameter. The instance-wise contrastive loss from augmentation t_2 $\ell_i^{t_2}$ can be formulated similarly. Then the average representation loss is formulated as

$$\mathcal{L}_{rep} = \frac{1}{2N}\sum_{i=1}^{N}\left(\ell_i^{t_1} + \ell_i^{t_2}\right), \tag{4}$$

For the augmented sample $x_i^{t_1}$, the cluster-wise feature matrix is constituted as

$$V^{t_1} = \begin{bmatrix} v_1^{t_1} \\ v_2^{t_1} \\ \vdots \\ v_N^{t_1} \end{bmatrix} \in R^{N\times K}, \tag{5}$$

where K is the number of cluster, $v_i^{t_1} = \varphi_C\left(NN\left(\xi_i^{t_1}, S\right)\right)$. The vector $v_i^{t_1}$ can be regarded as the probability of $x_i^{t_1}$ belonging to different clusters. Let $\tilde{v}_i^{t_1}$ be the ith column of V^{t_1}, which is the distribution of N samples in the ith cluster and can be also regarded as the feature representation of ith cluster. For the augmented sample $x_i^{t_2}$, the cluster-wise feature matrix is constituted as

$$V^{t_2} = \begin{bmatrix} v_1^{t_2} \\ v_2^{t_2} \\ \vdots \\ v_N^{t_2} \end{bmatrix} \in R^{N\times K}, \tag{6}$$

where $v_i^{t_2} = \varphi_C\left(\xi_i^{t_2}\right)$. Let $\tilde{v}_i^{t_2}$ be the ith column of V^{t_2}. The cluster-wise contrastive loss from augmentation t_1 is formulated as

$$\tilde{\ell}_i^{t_1} = -\log \frac{\exp\left(s\left(\tilde{v}_i^{t_1\top}, \tilde{v}_i^{t_2\top}\right)/\tau_C\right)}{\sum_{j=1}^{K}\mathbf{1}_{[j\neq i]}\left[\exp\left(s\left(\tilde{v}_i^{t_1\top}, \tilde{v}_j^{t_1\top}\right)/\tau_C\right) + \exp\left(s\left(\tilde{v}_i^{t_1\top}, \tilde{v}_j^{t_2\top}\right)/\tau_C\right)\right]}, \tag{7}$$

Algorithm 1: CCES algorithm

Input: dataset D; training epoch E; batch size N; temperature τ_R and τ_C; number of clusters K; backbone network: $\pi_\eta(\cdot)$; the initial support set S; MLP: $\varphi_R(\cdot)$ and $\varphi_C(\cdot)$.
Output: clustering index.

```
// Start training
for epoch=1:E do
    for x_i ∈ {D} do
        Randomly select {x_i}_{i=1}^N from dataset D to construct a minibatch.
        Obtain augmentations of x_i with ContrastiveCrop:
        x_i^{t1} = Conaug^{t1}(x_i), x_i^{t2} = Conaug^{t2}(x_i)
        Extract feature representations:
        ξ_i^{t1} = π_η(x_i^{t1}), ξ_i^{t2} = π_η(x_i^{t2})
        Obtain the nearest neighbor of ξ_i^{t1} in the support set S:
        NN(ξ_i^{t1}, S)
        Compute instance-wise representations:
        φ_R(NN(ξ_i^{t1}, S)), u_i^{t2} = φ_R(ξ_i^{t2})
        Generate the instance-wise contrastive loss L_rep by Eq. (4).
        Compute cluster-wise representations:
        v_i^{t1} = φ_C(NN(ξ_i^{t1}, S)), v_i^{t2} = φ_C(ξ_i^{t2})
        Generate the cluster-wise contrastive loss L_clu by Eq. (8).
        Update π_η(·), φ_R(·) and φ_C(·) by minimizing L.
        Update S by the first n (batch size) elements.
    end
end
// Start testing
for a test sample x do
    ξ = π_η(x) // generate representations
    μ = arg max φ_C(ξ) // generate clustering index
end
```

where τ_C is the cluster-wise temperature parameter. Similar to $\tilde{\ell}_i^{t_1}$, the cluster-wise contrastive loss from augmentation t_2 $\tilde{\ell}_i^{t_2}$ can be formulated, and the average cluster-wise loss is defined as

$$\mathcal{L}_{clu} = \frac{1}{2K}\sum_{i=1}^{K}\left(\tilde{\ell}_i^{t_1} + \tilde{\ell}_i^{t_2}\right) - H(Y), \tag{8}$$

In Eq. (8), the entropy $H(Y)$ is used to reduce the situation of trivial solutions in the training process, which is formulated as

$$H(Y) = -\sum_{i=1}^{K}\left[P\left(\tilde{v}_i^{t_1}\right)\log P\left(\tilde{v}_i^{t_1}\right) + P\left(\tilde{v}_i^{t_2}\right)\log P\left(\tilde{v}_i^{t_2}\right)\right], \tag{9}$$

where $P\left(\tilde{v}_i^{t}\right) = \frac{1}{N}\sum_{n=1}^{N} V_{ni}^{t}/\|V^t\|_1$, V_{ni}^{t} is the nth row and ith column of V^t, and $t \in \{t_1, t_2\}$.

CCES minimizes the instance-wise and cluster-wise contrastive loss simultaneously and the total loss is defined as

$$\mathcal{L} = \mathcal{L}_{rep} + \mathcal{L}_{clu} \tag{10}$$

The training and test process of the CCES is briefly shown in Algorithm 1.

4. Experiment

4.1. Datasets and evaluation metrics

We evaluate the proposed CCES through experiments on four challenging benchmark datasets, including CIFAR-10 [35], CIFAR-100 [35],

Table 1
Properties of four datasets.

Dataset	Image size	Train	Test	Classes
CIFAR-10	32 × 32	50,000	10,000	10
CIFAR-100	32 × 32	50,000	10,000	20
Tiny-ImageNet	64 × 64	100,000	10,000	200
STL-10	96 × 96	50,000	80,000	10

Tiny-ImageNet [36] and STL10 [37]. The four datasets is briefly described as follows.

CIFAR-10 [35]: There are 60,000 samples belonging to 10 categories and the size of each sample is 32 × 32. In our experiments, 50,000 samples are used for training and 10,000 samples for testing.

CIFAR-100 [35]: The number and the size of samples are the same as CIFAR-10. The samples are divided into 100 categories. To be consistent with previous works, we adopt the 20 super-classes as the ground-truth.

Tiny-ImageNet [36]: It is a subset of ImageNet with 200 categories. There are 100,000/10,000 training/testing samples evenly distributed in each category. The size of each sample is 64 × 64.

STL-10 [37]: It provides 13,000 samples with the size of 96 × 96. The samples are divided into 10 categories.

The statistics of four datasets is shown in Table 1.

We exploit three clustering metrics (ACC, NMI, and ARI) to evaluate the effectiveness of CCES. ACC is clustering accuracy, which is defined as

$$\mathrm{ACC}(\ell, C) = \max_{M} \frac{1}{n} \sum_{i=1}^{n} 1\left\{\ell_i = M\left(c_i\right)\right\}, \tag{11}$$

where ℓ_i is the ground truth, c_i is the predictive class and $M(\cdot)$ is the Hungarian mapping method.

NMI is normalized mutual information, which is defined as

$$\mathrm{NMI}(\ell, C) = 2 \cdot \frac{I(\ell; C)}{H(C) + H(\ell)}, \tag{12}$$

where $I(\ell; C)$ is the mutual information between ℓ and C, and $H(\cdot)$ denotes the entropy.

ARI is adjusted rand index, which is defined as

$$\mathrm{ARI} = \frac{RI - E[RI]}{\max(RI) - E[RI]}, \tag{13}$$

where RI denotes the similarity between the predicted value and the real value, and E denotes the expected values.

4.2. Implementation details

In CCES, all samples are resized to the size of 224 × 224. We employ ResNet [38] as the backbone network to extract 512-dimensional feature representations. In the beginning, the support set is initialized as a random matrix with the size of $h \times d$, where $h = 65536$ is the length of the support set and $d = 512$ is dimension of representations. For the instance-wise MLP $\varphi_R(\cdot)$, the output layer is set as 128 dimensions. For the cluster-wise MLP $\varphi_C(\cdot)$, the last layer is used to generate soft labels, whose dimensionality is equal to the number of categories. The temperature parameters and batch size are fixed on each dataset, which will be analyzed through ablation studies in Section 4.4. Nvidia GeForce RTX 3090 48G is adopted for 1200 epochs in all experiments. The learning rate of Adam optimizer is set as 10^{-4}.

4.3. Experimental results and analysis

Table 2 lists the clustering performances of CCES and other well-known clustering methods. To increase the diversity of baseline models and verify the effectiveness of CESS sufficiently, we choose three different kinds of clustering methods, including the traditional clustering methods K-means [1], SC [2], AC [3] and NMF [39], the recently popular deep clustering methods AE [40], SDAE [41], DECNN [42], JULE [6], DEC [5], DAC [43], IIC [30], DSEC [44] and PICA [8], and the latest contrastive clustering methods DCDC [13], CC [11], DRC [14], ConCURL [45] and CLKNN [17]. Benefiting from the effective augmentations and the novel way of constructing sample pairs, CCES basically performs better than other methods under three clustering metrics.

By leveraging the powerful feature extraction capabilities enabled by contrastive learning, CCES significantly outperforms traditional clustering methods and general deep clustering methods. Taking ACC as example, CCES is 58.3% higher than K-means on CIFAR-10 and 31.2% higher on CIFAR-100. Besides, compared with the general deep clustering algorithm, for ACC, CCES has 51.1% higher than DEC on CIFAR-10 and 25.7% higher on CIFAR-100. Compared with the contrastive clustering methods, CCES also performs better. It may be caused by that CCES provides the diversity of positive sample pair, which cannot be obtained via different augmentations. For instance, CCES is 8.5% higher than DRC for ACC, 10.3% for NMI, and 14.7% for ARI on CIFAR-10. Experimental result on CIFAR-100, STL-10, and Tiny-ImageNet are similar as well. From Table 2, we can also find that the ACC of CCES is lower than that of CC on STL-10 dataset. It is probably caused by the attribute of the images. On this dataset, most of the images are already in the central position, which can easily reduce the occurrence of false positives. However, on this dataset, CCES has higher NMI and ARI than CC. We suppose that the positive sample pair constructed through nearest-neighbor mining plays a crucial role, which avoids the neglect of semantic information in previous end-to-end contrastive clustering frameworks. With the nearest-neighbor mining, CCES can learn more discriminative features that are invariant to intra-class variations.

Further, Friedman–Nemenyi statistical test is utilized to check if the differences between CESS and other methods are significant. We first perform Friedman test. The null hypothesis is set to assume no difference between CCES and other methods. All the twenty methods ($k = 20$) and four datasets ($N = 4$) are adopted in the Friedman test and the significance level is set as $\alpha = 0.05$. Then for the metric ACC, we have Friedman static $\chi_F^2 = \frac{12N}{k(k+1)}\left[\sum_{i=1}^{k} \gamma_i^2 - \frac{k(k+1)^2}{4}\right] = 58.085$ and $F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2} = 9.727$, where γ_i is mean rank. Besides, we have the Critical Value $F_\alpha(k-1,(k-1)(N-1)) = F_{0.05}(19,57) = 1.772$. Since $F_F > F_{0.05}(19,57)$, we can reject the null hypothesis, and say that, for ACC, CCES has differences with other methods. We also perform Friedman test for the metrics NMI and ARI, and the results are listed in Table 3. From Table 3, we can see that $F_F = 14.382 > 1.772$ for NMI and $F_F = 13.716 > 1.772$ for ARI, indicating that, for NMI and ARI, CCES also has differences with other methods. Through Friedman test, we already know that CCES differs from other methods. Now we implement Nemenyi test to illustrate the degree of significance between CCES and other methods. We also set the significance level as 0.05. The critical difference (CD) value is calculated as $\mathrm{CD} = q_\alpha\sqrt{\frac{k(k+1)}{6N}} = 3.59$, where $q_\alpha = 2.84$. If the mean rank difference between two methods is greater than the CD value, then there is a significant difference between them, otherwise there is no significant difference (solid line is connected in the figure). The mean rank differences are shown in Fig. 4. From Fig. 4, we can find that CCES is significantly different from all the other methods except CC. CCES differs from CC but the difference is not significant, due to the fact that CC performs stably and ranks well for all the three metrics on four datasets. However, with more effective sample pairs, the proposed CCES still outperforms CC on different datasets.

4.4. Ablation study

In this section, we carry out ablation studies onCIFAR-10, CIFAR-100 and Tiny-ImageNet datasets respectively to verify the effectiveness of ContrastiveCrop and nearest-neighbor mining.

Results of ablation experiments on different datasets are shown in Table 4, where RCrop represents RandomCrop for data augmentation,

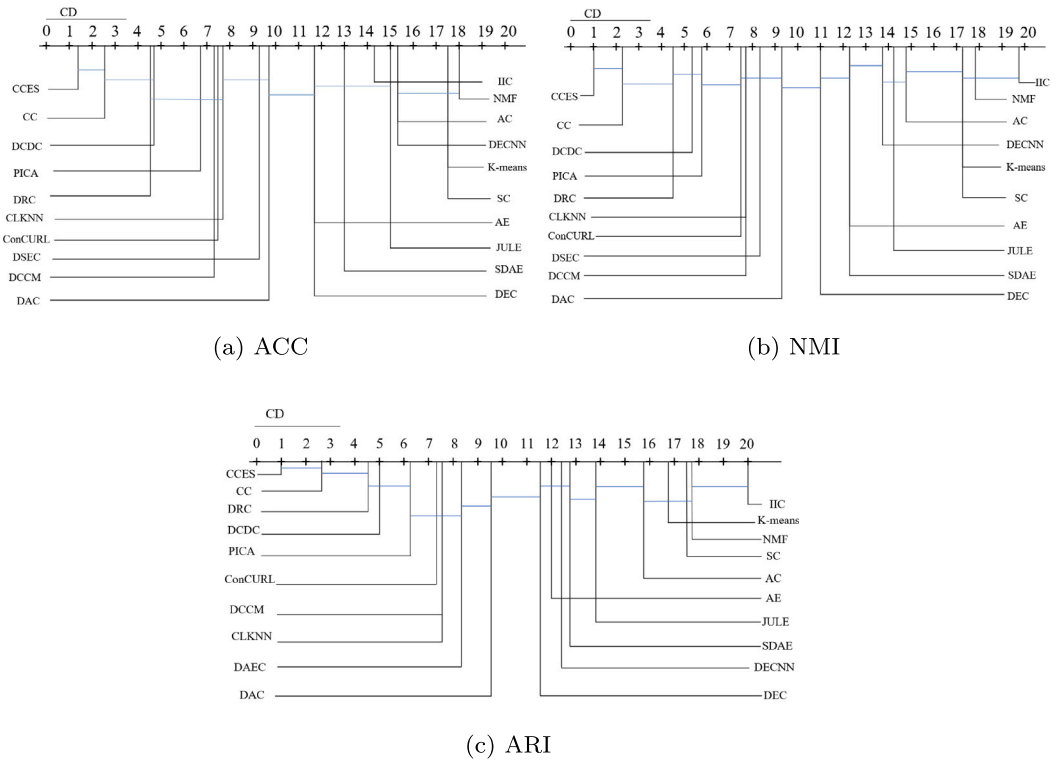

Fig. 4. Nemenyi test results of different evaluation metrics.

Table 2
The clustering performance on four challenging benchmarks.

Datasets	CIFAR-10			CIFAR-100			Tiny-ImageNet			STL-10		
Metrics	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
K-means	0.229	0.087	0.049	0.130	0.084	0.028	0.025	0.065	0.005	0.192	0.125	0.061
SC	0.247	0.103	0.085	0.136	0.090	0.022	0.022	0.063	0.004	0.159	0.098	0.048
AC	0.228	0.108	0.065	0.138	0.098	0.034	0.027	0.069	0.005	0.332	0.239	0.140
NMF	0.190	0.081	0.034	0.118	0.079	0.026	0.029	0.072	0.005	0.180	0.096	0.046
AE	0.314	0.239	0.169	0.165	0.100	0.048	0.041	0.131	0.007	0.303	0.250	0.161
SDAE	0.297	0.251	0.163	0.151	0.111	0.046	0.039	0.127	0.007	0.302	0.224	0.152
DECNN	0.282	0.240	0.174	0.133	0.092	0.038	0.032	0.111	0.006	0.299	0.227	0.162
JULE	0.272	0.192	0.138	0.137	0.103	0.033	0.033	0.102	0.006	0.277	0.182	0.164
DEC	0.301	0.257	0.161	0.185	0.136	0.050	0.037	0.115	0.070	0.359	0.276	0.186
DAC	0.522	0.396	0.306	0.238	0.185	0.088	0.066	0.190	0.017	0.470	0.366	0.257
DCCM	0.623	0.496	0.408	0.327	0.285	0.173	0.108	0.224	0.038	0.482	0.376	0.262
IIC	0.617	N/A	N/A	0.257	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
DSEC	0.478	0.438	0.340	0.255	0.212	0.110	0.066	0.190	0.017	0.482	0.403	0.286
PICA	0.696	0.591	0.512	0.337	0.310	0.171	0.098	0.277	0.040	0.713	0.611	0.531
DCDC	0.699	0.585	0.506	0.349	0.310	0.179	0.164	0.323	0.073	0.734	0.621	0.547
CLKNN	0.803	0.686	0.638	0.407	0.418	0.251	N/A	N/A	N/A	0.716	0.596	0.519
CC	0.760	0.682	0.607	0.426	0.428	0.266	0.142	0.340	0.071	**0.854**	0.738	0.719
DRC	0.727	0.621	0.547	0.367	0.356	0.208	0.139	0.321	0.056	0.747	0.644	0.569
ConCURL	0.785	0.667	0.614	0.409	0.390	0.232	N/A	N/A	N/A	0.752	0.645	0.580
CCES (Ours)	**0.812**	**0.724**	**0.694**	**0.442**	**0.436**	**0.301**	**0.196**	**0.382**	**0.125**	0.847	**0.775**	**0.731**

The best results have been highlighted in bold. "N/A" denotes that the value is unavailable.

Table 3
The statistical F_F value of each metric and critical value in the Friedman test.

Metric	F_F	Critical value ($\alpha = 0.05$)
ACC	9.727	1.772
NMI	14.382	
ARI	13.716	

CCrop represents ContastiveCrop for data augmentation, NN(single1) and NN(single2) represent the nearest-neighbor mining only for instance-wise and cluster-wise contrastive learning respectively, and NN(double) represents the nearest-neighbor mining for both instance-wise and cluster-wise contrastive learning.

From Table 4, we can see that, on the whole, the performance of CCrop is better than RCrop. In the case of CCrop, combining with NN can obtain better results. It can be also seen that NN(double) is more effective than NN(single1) and NN(single2). NN(double) can introduce more abundant semantic information than single nearest-neighbor mining manners. Besides, the performance of NN(single1) and NN(single2) is basically the same. It indicates that constructing positive sample pairs by a sample and its nearest neighbors sufficiently considers the information of different samples belonging to the same cluster, providing more semantic variations and making positive sample pairs

Table 4
Results of ablation experiments.

Dataset	RCrop	CCrop	NN(single1)	NN(single2)	NN(double)	ACC	NMI	ARI
CIFAR-10	✓					76.0%	68.2%	60.7%
		✓				76.4%	68.6%	61.2%
	✓		✓			79.2%	69.6%	62.5%
	✓			✓		79.5%	71.2%	67.9%
	✓				✓	80.6%	71.9%	68.8%
		✓	✓			79.8%	72.2%	69.0%
		✓		✓		80.3%	71.7%	68.4%
		✓			✓	**81.2%**	**72.4%**	**69.4%**
CIFAR-100	✓					41.8%	42.2%	26.6%
		✓				42.6%	42.7%	26.5%
	✓		✓			43.3%	43.1%	27.9%
	✓			✓		43.0%	42.9%	27.8%
	✓				✓	43.9%	43.1%	29.1%
		✓	✓			43.5%	42.8%	28.3%
		✓		✓		43.7%	42.6%	28.7%
		✓			✓	**44.2%**	**43.6%**	**30.1%**
Tiny-ImageNet	✓					14.2%	34.0%	7.1%
		✓				15.3%	35.1%	8.4%
	✓		✓			16.6%	36.4%	9.5%
	✓			✓		16.2%	35.9%	9.2%
	✓				✓	18.9%	37.2%	11.2%
		✓	✓			18.3%	36.8%	10.5%
		✓		✓		16.7%	36.6%	10.1%
		✓			✓	**19.6%**	**38.2%**	**12.5%**

"✓" denotes that we select the corresponding component.

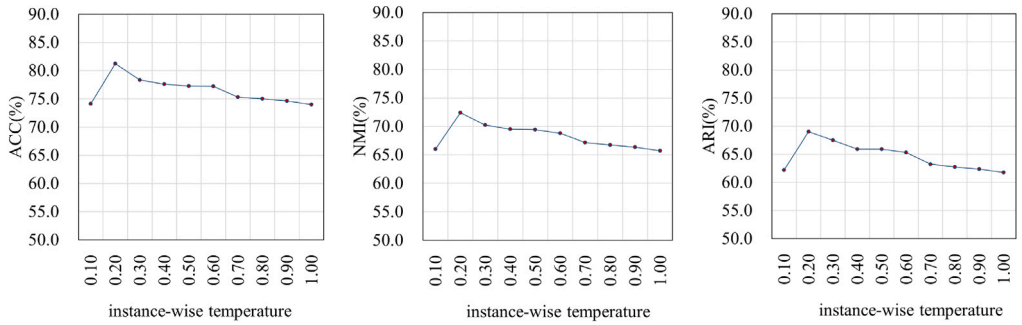

Fig. 5. The values of ACC (%), NMI (%) and ARI (%) with different instance-wise temperature parameters on CIFAR-10 dataset.

more diverse. In addition, NN can bring more obvious performance improvement than CCrop. It indicates that CCrop may only provide support for sample processing before feature extraction, while constructing positive sample pairs by NN is able to extract more discriminative feature representations for instance-wise and cluster-wise contrastive learning.

4.5. Temperature parameters study

In this section, we conduct temperature parameter experiments on CIFAR-10 dataset. We empirically let the instance-wise and cluster-wise temperature parameters vary from 0.10 to 1.00, with the interval of 0.1. Figs. 5 and 6 depict the performance of CCES with different instance-wise and cluster-wise temperature parameters on CIFAR-10 dataset. From these figures, we have the following findings: when the instance-wise temperature parameter τ_R is set as 0.20 and the cluster-wise temperature parameter τ_C is set as 1.00, the superior performance can be achieved on CIFAR-10 dataset. Based on this, the temperature parameter τ_R is fixed at 0.20 and τ_C is fixed at 1.00 in all our experiments.

4.6. Convergence analysis

In this section, we perform the convergence experiments on CIFAR-10 and Tiny-ImageNet datasets. Figs. 7 and 8 depict the variation of ACC, NMI and ARI versus different epochs. From Figs. 7 and 8, we can find that the values of ACC, NMI and ARI all increase with the increase of epochs. After about 1000 epochs, the curves of values of three metrics tend to be flat. The loss of CCES versus different epochs is depicted in Fig. 9. From Fig. 9, we can see that the loss rapidly decreases in the first 300 epochs and the loss curve tends to be flat after 1000 epochs.

5. Conclusion

In this paper, we present an end-to-end contrastive clustering method CCES. CCES utilizes ContrastiveCrop and nearest-neighbor mining simultaneously and achieves the promising performance. ContrastiveCrop is used for data augmentation, which can reduce the occurrence of false positives. Nearest-neighbor mining is employed to construct positive sample pairs, which makes a sample and its nearest neighbor as positive sample pairs. Extensive experiments on four challenging benchmarks verify the superiority of CCES. Through ablations experiments, we find that ContrastiveCrop and nearest-neighbor mining are both useful for improving the clustering performance of CCES. The support set is the key to performing the nearest neighbor mining. To improve the performance of CCES, in the future, we will try to build the support set by randomly selecting the representations of the samples in the dataset, which may extract more semantic features. In addition,

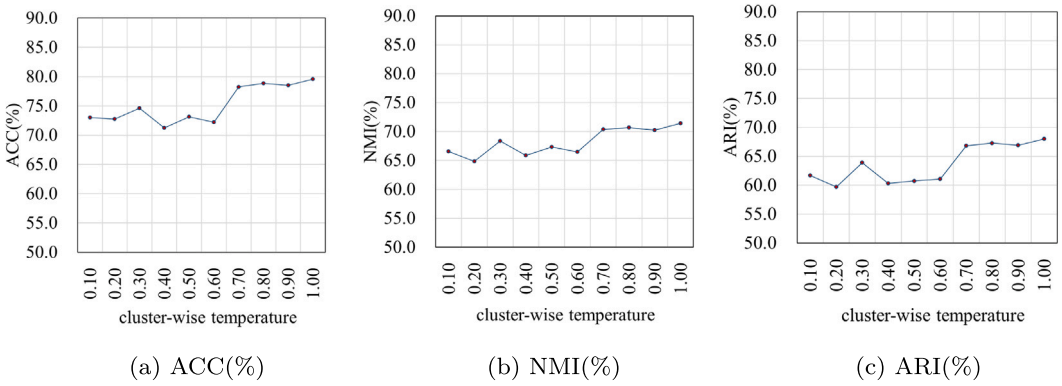

Fig. 6. The values of ACC (%), NMI (%) and ARI (%) with different cluster-wise temperature parameters on CIFAR-10 dataset.

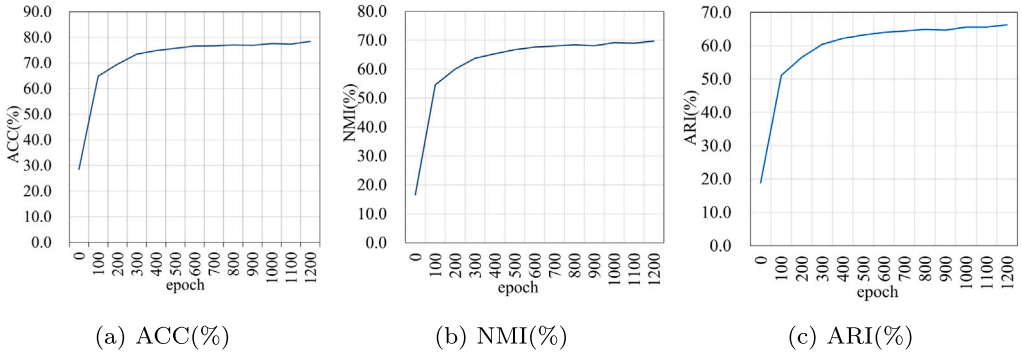

Fig. 7. The values of ACC (%), NMI (%) and ARI (%) vary with different epochs on CIFAR-10 dataset.

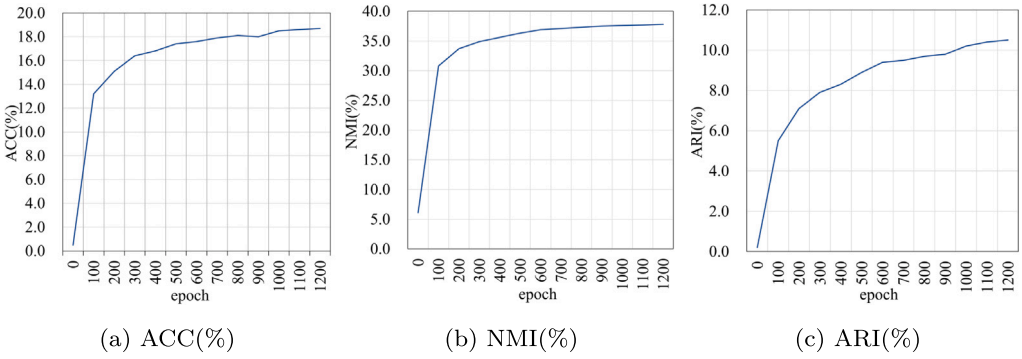

Fig. 8. The values of ACC (%), NMI (%) and ARI (%) vary with different epochs on Tiny-ImageNet dataset.

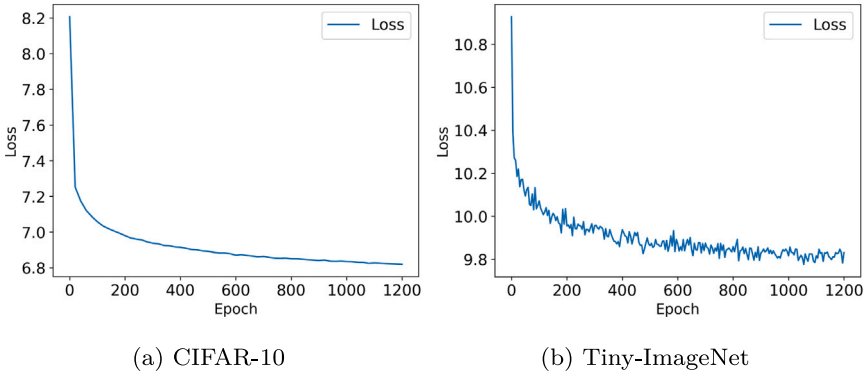

Fig. 9. Loss curves vary with different epochs on CIFAR-10 and Tiny-ImageNet datasets.

since CCES is an unsupervised learning method, two samples belonging to the same category are probably regarded as negative samples pairs. To alleviate this problem, in the future, we plan to extend CCES to semi-supervised learning.

CRediT authorship contribution statement

Jun Yin: Conceptualization, Methodology, Software, Writing – review & editing. **Haowei Wu:** Data curation, Writing – original draft. **Shiliang Sun:** Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported by Shanghai Pujiang Program (Grant No. 22PJD029), National Natural Science Foundation of China (Grants No. 62076096), Shanghai Municipal Project (Grant No. 2051110090), KLATASDS-MOE and the Fundamental Research Funds for the Central Universities.

References

[1] J.B. MacQueen, Some methods for classification and analysis of multivariate observations, in: Berkeley Symposium on Mathematical Statistics and Probability, vol. 14, 1966, pp. 281–297.
[2] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in: Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic, 2001, pp. 849–856.
[3] P. Franti, O. Virmajoki, V. Hautamaki, Fast agglomerative clustering using a k-nearest neighbor graph, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 1875–1881.
[4] Y. Lecun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436.
[5] J. Xie, R. Girshick, A. Farhadi, Unsupervised deep embedding for clustering analysis, in: International Conference on Machine Learning, 2016, pp. 478–487.
[6] J. Yang, D. Parikh, D. Batra, Joint unsupervised learning of deep representations and image clusters, in: 2016 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 5147–5156.
[7] M. Caron, P. Bojanowski, A. Joulin, M. Douze, Deep clustering for unsupervised learning of visual features, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 132–149.
[8] J. Huang, S. Gong, X. Zhu, Deep semantic clustering by partition confidence maximisation, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020, pp. 8849–8858.
[9] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: International Conference on Machine Learning, PMLR, 2020, pp. 1597–1607.
[10] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020, pp. 9729–9738.
[11] Y. Li, P. Hu, Z. Liu, D. Peng, J.T. Zhou, X. Peng, Contrastive clustering, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, 2021, pp. 8547–8555.
[12] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, L. Van Gool, Scan: Learning to classify images without labels, in: Proceedings of the European Conference on Computer Vision, ECCV, Springer, 2020, pp. 268–285.
[13] Z. Dang, C. Deng, X. Yang, H. Huang, Doubly contrastive deep clustering, 2021, arXiv Preprint arXiv:2103.05484.
[14] H. Zhong, C. Chen, Z. Jin, X.-S. Hua, Deep robust clustering by contrastive learning, 2020, arXiv Preprint arXiv:2008.03030.
[15] X. Peng, K. Wang, Z. Zhu, M. Wang, Y. You, Crafting better contrastive views for siamese representation learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 16031–16040.
[16] D. Dwibedi, Y. Aytar, J. Tompson, P. Sermanet, A. Zisserman, With a little help from my friends: Nearest-neighbor contrastive learning of visual representations, in: International Conference on Computer Vision, 2021, pp. 9568–9577.
[17] X. Zhang, S. Wang, Z. Wu, X. Tan, Unsupervised image clustering algorithm based on contrastive learning and K-nearest neighbors, Int. J. Mach. Learn. Cybern. (2022) 1–9.
[18] R. Hadsell, S. Chopra, Y. LeCun, Dimensionality reduction by learning an invariant mapping, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR'06, vol. 2, IEEE, 2006, pp. 1735–1742.
[19] M. Gutmann, A. Hyvärinen, Noise-contrastive estimation: A new estimation principle for unnormalized statistical models, in: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, vol. 9, 2010, pp. 297–304.
[20] A.v.d. Oord, Y. Li, O. Vinyals, Representation learning with contrastive predictive coding, 2018, arXiv Preprint arXiv:1807.03748.
[21] X. Wang, G.-J. Qi, Contrastive learning with stronger augmentations, IEEE Trans. Pattern Anal. Mach. Intell. (2022).
[22] M. Zheng, S. You, F. Wang, C. Qian, C. Zhang, X. Wang, C. Xu, Ressl: Relational self-supervised learning with weak augmentation, Adv. Neural Inf. Process. Syst. 34 (2021) 2543–2555.
[23] J.D. Robinson, C.-Y. Chuang, S. Sra, S. Jegelka, Contrastive learning with hard negative samples, in: International Conference on Learning Representations, 2020.
[24] Y. Kalantidis, M.B. Sariyildiz, N. Pion, P. Weinzaepfel, D. Larlus, Hard negative mixing for contrastive learning, Adv. Neural Inf. Process. Syst. 33 (2020) 21798–21809.
[25] R. Zhu, B. Zhao, J. Liu, Z. Sun, C.W. Chen, Improving contrastive learning by visualizing feature transformation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10306–10315.
[26] P. Ji, T. Zhang, H. Li, M. Salzmann, I. Reid, Deep subspace clustering networks, Adv. Neural Inf. Process. Syst. 30 (2017).
[27] Z. Jiang, Y. Zheng, H. Tan, B. Tang, H. Zhou, Variational deep embedding: An unsupervised and generative approach to clustering, in: International Joint Conference on Artificial Intelligence, 2017, pp. 1965–1972.
[28] J. Wu, K. Long, F. Wang, C. Qian, C. Li, Z. Lin, H. Zha, Deep comprehensive correlation mining for image clustering, in: 2019 IEEE/CVF International Conference on Compute Vision, ICCV 2019, 2019, pp. 8149–8158.
[29] X. Guo, X. Liu, E. Zhu, X. Zhu, M. Li, X. Xu, J. Yin, Adaptive self-paced deep clustering with data augmentation, IEEE Trans. Knowl. Data Eng. 32 (2019) 1680–1693.
[30] X. Ji, A. Vedaldi, F.J. Henriques, Invariant information clustering for unsupervised image classification and segmentation, in: 2019 IEEE/CVF International Conference on Compute Vision, ICCV 2019, 2019, pp. 9864–9873.
[31] X. Deng, D. Huang, D.-H. Chen, C.-D. Wang, J.-H. Lai, Strongly augmented contrastive clustering, Pattern Recognit. (2023) 109470.
[32] Z. Dang, C. Deng, X. Yang, K. Wei, H. Huang, Nearest neighbor matching for deep clustering, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13693–13702.
[33] C. Xu, R. Lin, J. Cai, S. Wang, Deep image clustering by fusing contrastive learning and neighbor relation mining, Knowl.-Based Syst. 238 (2022) 107967.
[34] C. Niu, H. Shan, G. Wang, Spice: Semantic pseudo-labeling for image clustering, IEEE Trans. Image Process. 31 (2022) 7264–7278.
[35] A. Krizhevsky, G. Hinton, et al., Learning Multiple Layers of Features from Tiny Images, Department of Computer Science, University of Toronto, 2009.
[36] Y. Le, X. Yang, Tiny imagenet visual recognition challenge, in: CS 231N7, vol. 7, 2015, p. 3.
[37] A. Coates, A. Ng, H. Lee, An analysis of single-layer networks in unsupervised feature learning, in: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2011, pp. 215–223.
[38] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 770–778.
[39] C. Deng, X. He, X. Wang, H. Bao, J. Han, Locality preserving nonnegative matrix factorization, in: International Jont Conference on Artifical Intelligence, 2009.
[40] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Greedy layer-wise training of deep networks, Adv. Neural Inf. Process. Syst. 19 (2006).
[41] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, L. Bottou, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion., J. Mach. Learn. Res. 11 (2010).
[42] M.D. Zeiler, D. Krishnan, G.W. Taylor, R. Fergus, Deconvolutional networks, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2528—2535.
[43] J. Chang, L. Wang, G. Meng, S. Xiang, C. Pan, Deep adaptive image clustering, in: 2017 IEEE International Conference on Computer Vision, ICCV, 2017.
[44] J. Chang, G. Meng, L. Wang, S. Xiang, C. Pan, Deep self-evolution clustering, IEEE Trans. Pattern Anal. Mach. Intell. 42 (2018) 809–823.
[45] J.R. Regatti, A.A. Deshmukh, E. Manavoglu, U. Dogan, Consensus clustering with unsupervised representation learning, in: International Joint Conference on Neural Network, 2020.